RSA vs ECC: A COMPARATIVE ANALYSIS

Ujjwal Kumar Garg
Roll No. 21075153

Supervisor: Prof. P. P. Murthy

Department of Mathematics,
Guru Ghasidas Vishwavidalya

12 July, 2023

Discussion.

1. ECC
   - Definition
   - Algorithm
2. Analysis of RSA vs ECC
   - 8 bits – Encryption and Decryption
   - 64 bits – Encryption and Decryption
3. Advantage of ECC over RSA

Definition

- An elliptic curve is the set of points that satisfy a specific mathematical equation.
- The equation for an elliptic curve is $y^2 = x^3 + ax + b$
  also $4a^3 + 27b^2 \neq 0$

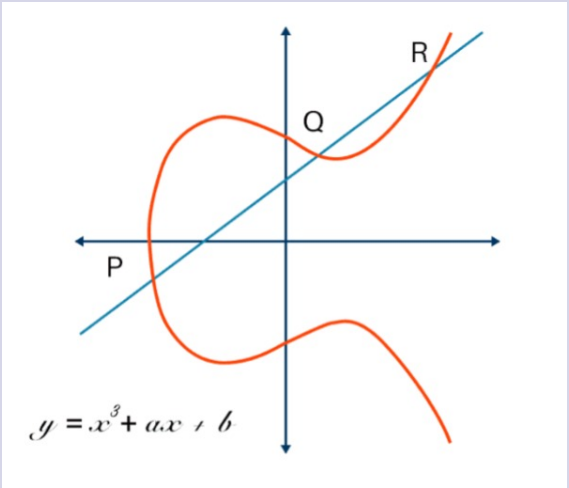


Figure:

ECC key exachange

Global Public Elements

$E_p(a, b)$: Elliptic curve with parameter a, b
p: prime number
G: point on curve whose order is large value of n

User A key genration

- Select private key $n_A, n_A < n$
- Calculate Public key $P_A = n_A * G$

User B key genration

- Select Private key $n_B, n_B < n$
- Calculate Public key $P_B = n_B * G$

Calculation of secret by user A

$K = n_A * P_B$

Calculation of secret key by user B

$K = n_B * P_A$

Encryption

- Let M the message
  encode M point on the elliptic curve. Let point be P_M
- For encryption chose a positve random integer k
- Cipher point $C_M = KG, P_M + kP_B$

Decryption

- Multiply the first point by the reciver private key $kG * n_B$
- $P_M + kP_B - kG * n_B = P_M$

Encrytion and Decryption

Global parameter of ECC

Here prime number $p = 11, a = 1, b = 1$ for encoding and decoding of message in ellptic curve. Based on global parameters, the ellptic curve equatons become:

$y^2 \mod 11 = (x^3 + x + 1) \mod 11$

Step: 1 Encoded a plain text message as a point on the curve

Lets consider the point to be encoded plain text message on the curve $M \in E_{11}(1, 1)$ is $(4, 6)$

Step: 2 Eastablish the Public key and Privte key

Chose a generator point $G \in E_{11}(1, 1)$, let G is (1,5)$\in E_{11}$(1,1)

Select a private key $n = 2$

Compute the Public key as $P_A = nG$

- Let nG equal to (x_3, y_3) as $n = 2$ and $G = (1, 5)$
- $P_A = 2G = G + G = (1, 5) + (1, 5)$
- Let $x_1 = x_2 = 1$ and $y_1 = y_2 = 5$
- $\lambda = \frac{3x^2+a}{2y} \mod 11 = \frac{3*1^2+1}{2*5} \mod 11 = 7$
- $x_3 = \lambda^2\text{-}x_1\text{-}x_2 \mod 11 = 7^2 - 1 - 1 \mod 11 = 3$
- $y_3 = \lambda(\text{a-}x_3)$ - y $\mod 11 = 7(1\text{-}3)$ - 5 $\mod 11 = 3$

now we have $(x_3, y_3) = (3,3)$

Step:3 Encrypt the message using Public key

$C = [kG, M + kP_A]$, where k is a random number

$C = [C_1, C_2]$

Let $k = 2$

$C = [2(1,5), (4,6) + 2(3,3)]$

$C = [(1,5) + (1,5), (4,6) + (3,3) + (3,3)]$

$C = [(3,3), (4,6) + (3,3) + (3,3)]$

$C = [(3,3), (4,6) + (6,5)]$

$C = [(3,3), (4,5)]$

$C_1 = (3,3)$ and $C_2 = (4,5)$

Step:4 Decrypt using private key

$M = C_2 - [nC_1]$

$M = (4, 5) - [2(3, 3)]$

$M = (4, 5) - [(3, 3), (3, 3)]$

$M = (4, 5) - (6, 5)$

$M = (4, 5) + (6, -5)$

$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \mod 11 = \frac{-5-5}{6-4} \mod 11 = \text{-5} \mod 11 = 6$

$x_3 = \lambda^2\text{-}x_1\text{-}x_2 \mod 11 = 6^2 - 4 - 6 \mod 11 = 26 \mod 11 = 4$

$y_3 = \lambda(x_1 - x_3) - y_1 \mod 11 = 6(4 - 4) - 5 \mod 11 = 6$

$(x_3, y_3) = (4, 6)$

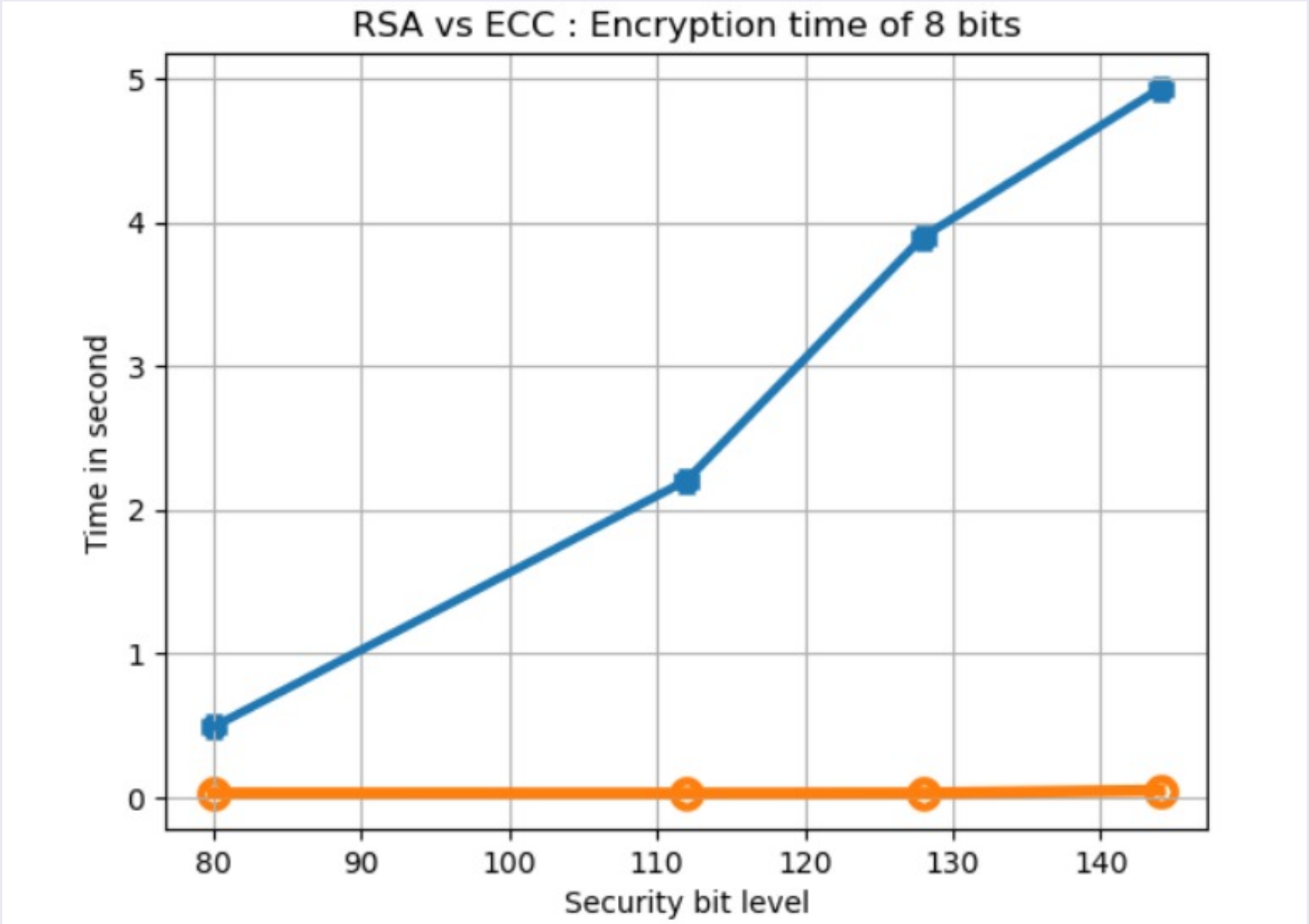
Analysis of RSA vs ECC

8 bits – Encryption and Decryption Time

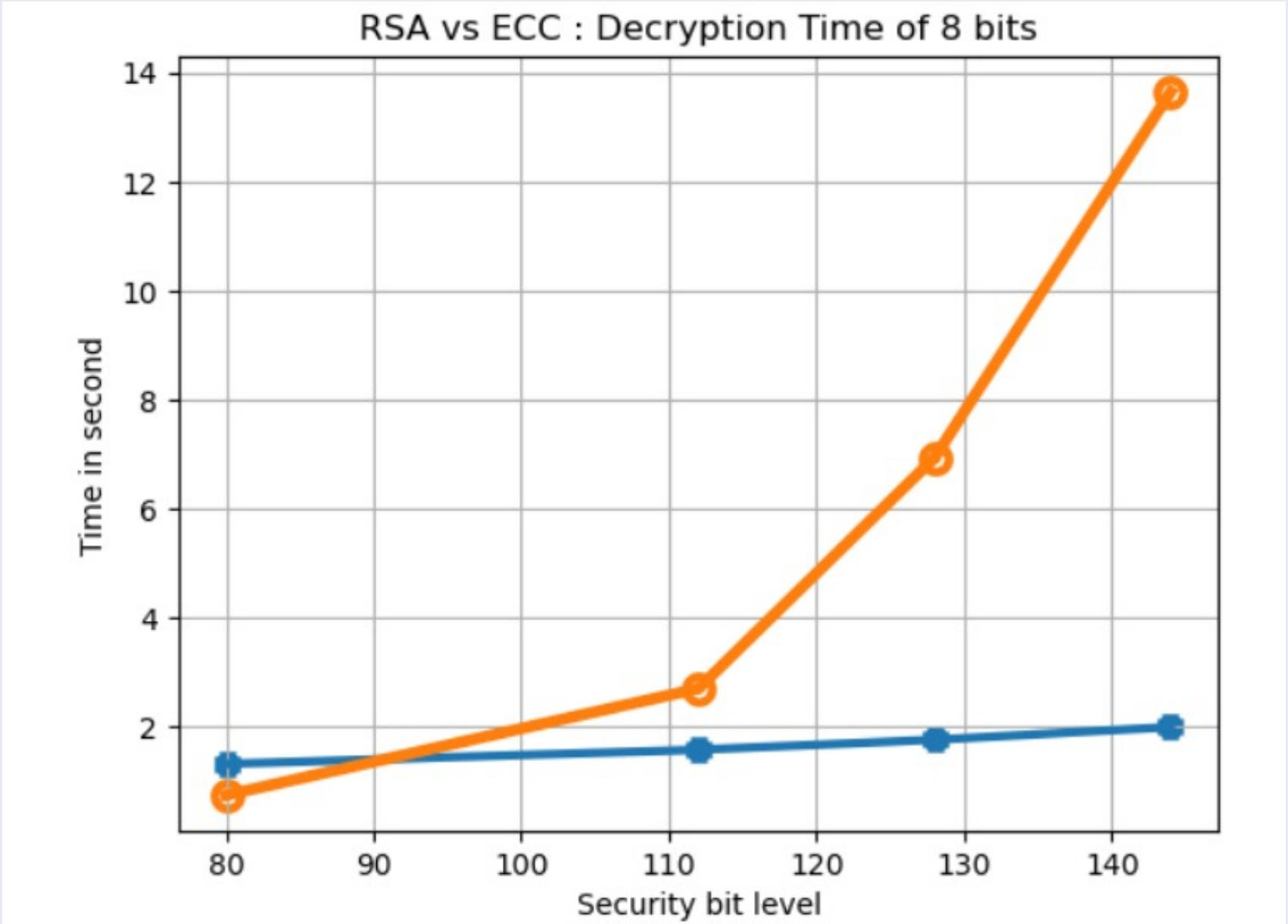
Security bit level	ECC Enc. Time	RSA Enc. Time	ECC Dec. Time	RSA Dec. Time
80	0.4885	0.0307	1.3267	0.7543
112	2.2030	0.0299	1.5863	2.7075
128	3.8763	0.0305	1.7690	6.9409
144	4.7266	0.0489	2.0022	13.6472

- $x = [80, 112, 128, 144]$
- $y = [.4885, 2.2030, 3.8963, 4.9266]$
- $z = [0.0307, 0.0299, 0.0305, 0.0489]$
- plt.plot(x,y , linewidth = 3)
- plt.plot(x,z , linewidth =4)
- plt.scatter(x,y , marker="+" , linewidth=9)
- plt.scatter(x,z , marker = "." ,linewidth=9)
- plt.title("RSA vs ECC : Encryption time of 8 bits")
- plt.xlabel("RSA vs ECC : Ecryption Time of 8 bits")
- plt.ylabel("Time in second")
- plt.grid("true")
- plt.show()

Blue = ECC , Ornage = RSA


RSA vs ECC : Encryption time of 8 bits
Time in second
0
1
2
3
4
5
80
90
100
110
120
130
140
Security bit level

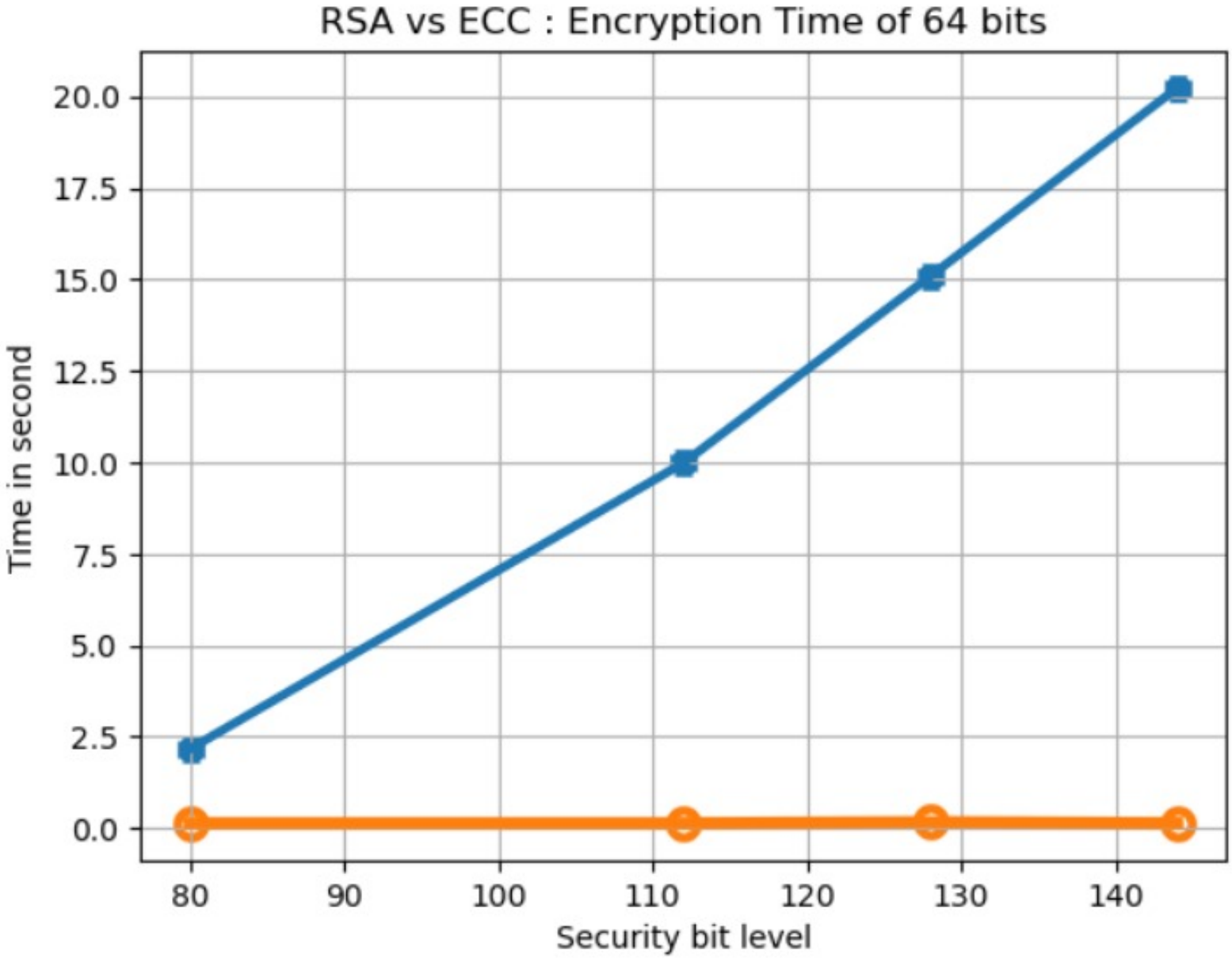
- x = [80,112,128,144]
- y1 = [1.3267,1.5863,1.7690,2.0022]
- z1 = [0.7543,2.7075,6.9409,13.6472]
- plt.plot(x,y1 , linewidth=3)
- plt.plot(x,z1 , linewidth=4)
- plt.scatter(x,y1,marker="+" , linewidth=9)
- plt.scatter(x,z1,marker=".",linewidth=9)
- plt.title("RSA vs ECC : Decryption Time of 8 bits")
- plt.xlabel("Security bit level")
- plt.ylabel("Time in second")
- plt.grid("true")
- plt.show()


RSA vs ECC : Decryption Time of 8 bits
Time in second
Security bit level
14
12
10
8
6
4
2
80
90
100
110
120
130
140

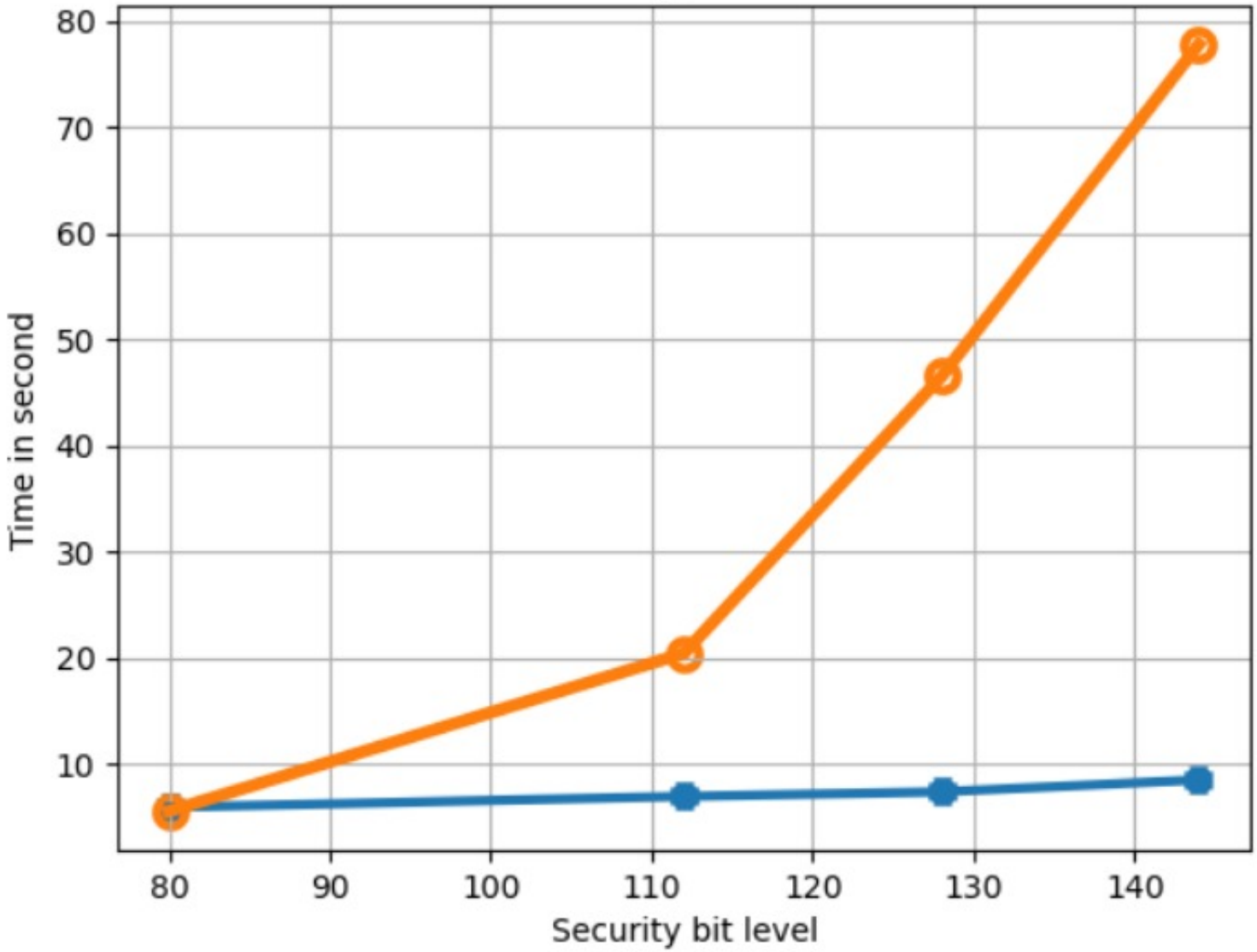
64 bits – Encryption and Decryption

Security bit level	ECC Enc. Time	RSA Enc. Time	ECC Dec. Time	RSA Dec. Time
80	2.1685	0.1366	5.9099	5.5372
112	9.9855	0.1635	6.9333	20.4108
128	15.0882	0.1672	7.3584	46.4782
144	20.2308	.01385	8.4785	77.7642

- x = [80,112,128,144]
- y2 = [2.1685 ,9.9855 ,15.0882 ,20.2308]
- z2= [0.1366 ,0.1366 ,0.1672 ,0.1385]
- plt.plot(x,y2 , linewidth=3)
- plt.plot(x,z2 , linewidth=4)
- plt.scatter(x,y2,marker="+" , linewidth=9)
- plt.scatter(x,z2,marker=".",linewidth=9)
- plt.title("RSA vs ECC : Encryption Time of 64 bits")
- plt.xlabel("Security bit level")
- plt.ylabel("Time in second")
- plt.grid("true")
- plt.show()


RSA vs ECC : Encryption Time of 64 bits
Time in second
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
80
90
100
110
120
130
140
Security bit level

- x = [80,112,128,144]
- y3 = [5.9099,6.9333,7.3584,8.4785]
- z3 = [5.5372,20.4108,46.4782,77.7642]
- plt.plot(x,y3 , linewidth=3)
- plt.plot(x,z3 , linewidth=4)
- plt.scatter(x,y3,marker="+" , linewidth=9)
- plt.scatter(x,z3,marker=".",linewidth=9)
- plt.title("RSA vs ECC : Decryption Time of 64 bits")
- plt.xlabel("Security bit level")
- plt.ylabel("Time in second")
- plt.grid("true")
- plt.show()


RSA vs ECC : Decryption Time of 64 bits
Time in second
Security bit level
80
70
60
50
40
30
20
10
80
90
100
110
120
130
140

Advantage of ECC over RSA

- ECC, it takes one−sixth the computational effort to provide the same level of cryptographic security that you get with 1024 bit RSA and is 15 time faster

Symmetric Encryption Key size in bits	RSA and DH key size	ECC key size
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

REFERENCES

- Batina, L., Mentens, N., Sakiyama, K., Preneel, B., and Verbauwhed, I., Public- Key Cryptography on the Top of Niddle, https://www.researchgate.net/publication/221381449, (2007), 1831-1834.
- Mahto, D., And Yadav, D.K., RSA and ECC : A comparative analysis, International Journal of applied engineering research, 12(19) (2017), 9053-9061.
- Mahto, D., And Yadav D. K., Performance Analysis of RSA and Elliptic Curve Cryptography, International Journal of Netw. Secur., 20(4) (2018), 625-635.
- Raju, GVS and Akbani, R., Elliptic curve cryptosystem and it's applications, SMC'03 Conference Proceedings (IEEE International Conference on Systems, Man and Cybernetics), 2 (2003), 1540-1543.


Vigila, S.M.C., and Muneeswaran, k., Implementation of text based cryptosystem using elliptic curve cryptography, First International Conference on Advanced Computing (IEEE), (2009), 82-85.


W.stallings, Cryptography and Network security, 5th ed. Boston: Prentice Hall,(2011)

Thanking You